

Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 7/3/2023

Name: Yaman Gupta	SRN: PES2UG21CS619	Section J
-------------------	-----------------------	--------------

Week# 7 Program Number: 1

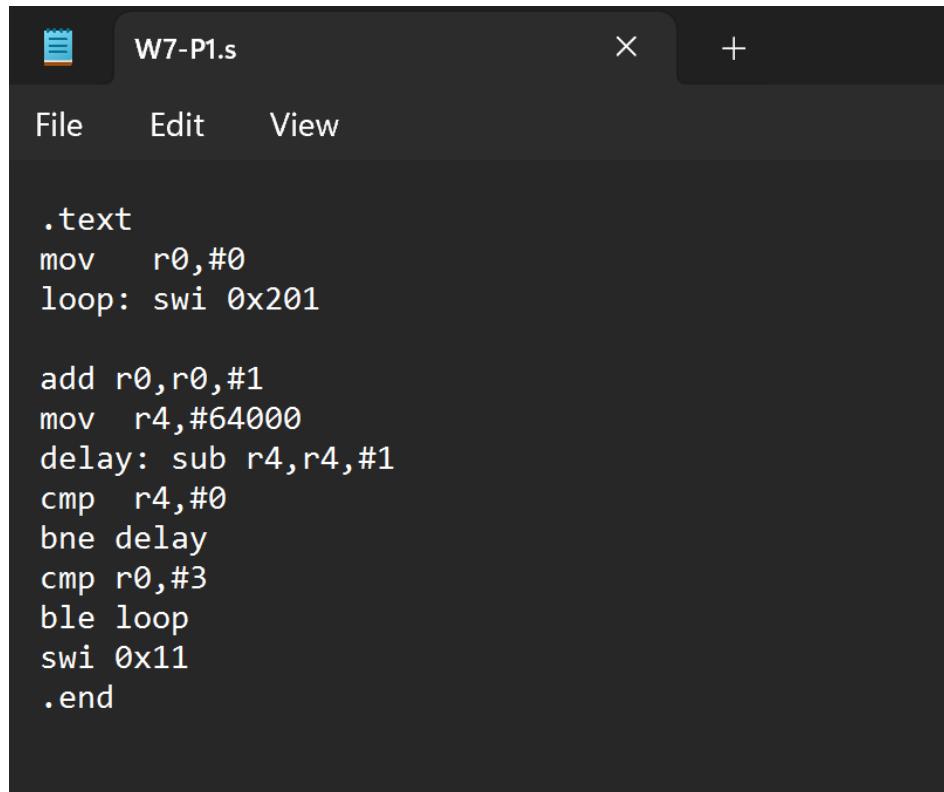
Title of the Program

**Write a Program to make the LED light in the Embest Board
Plugin Blink**

I.ARM Assembly Code

II. Output Screen Shots (Three)

I. ARM Assembly Code

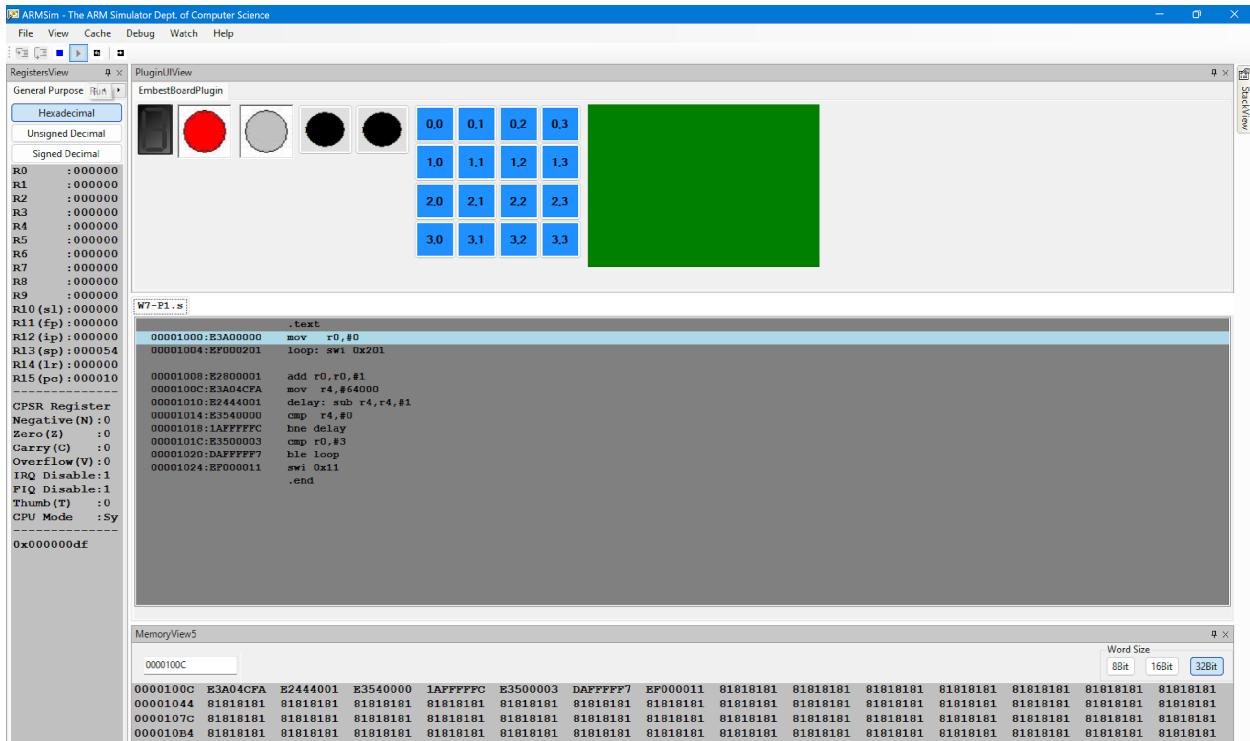
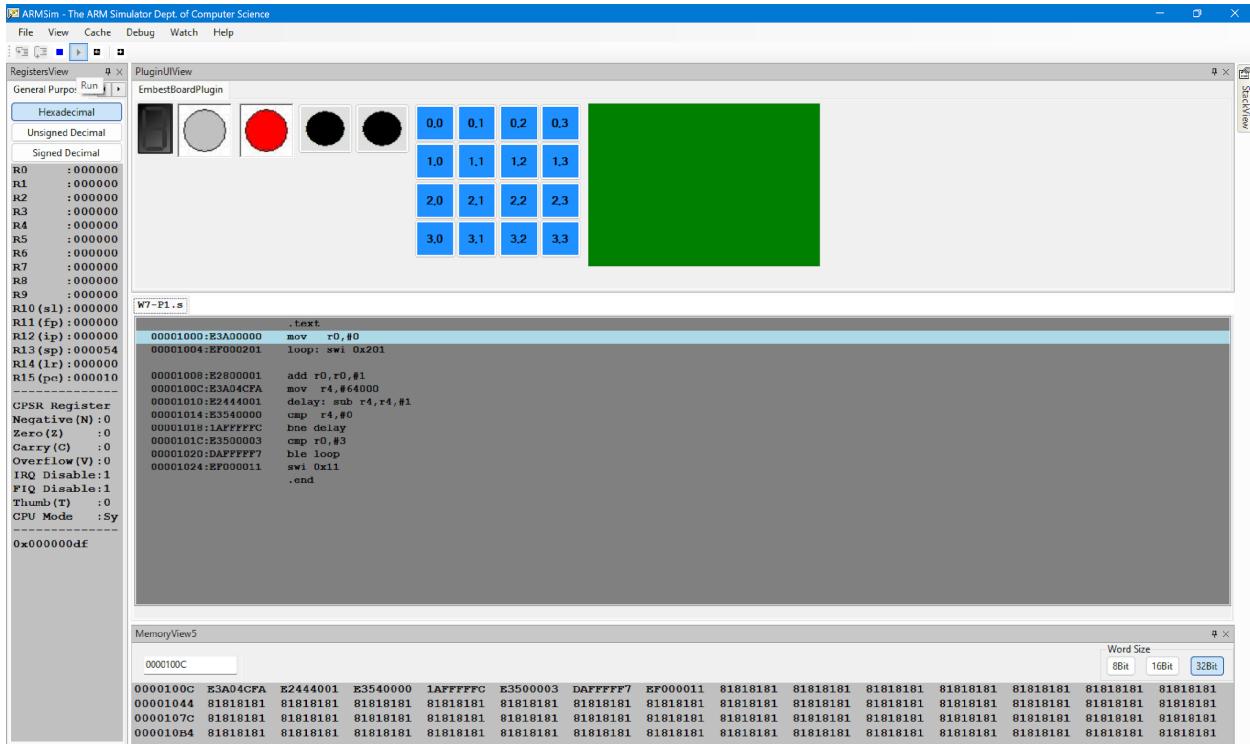


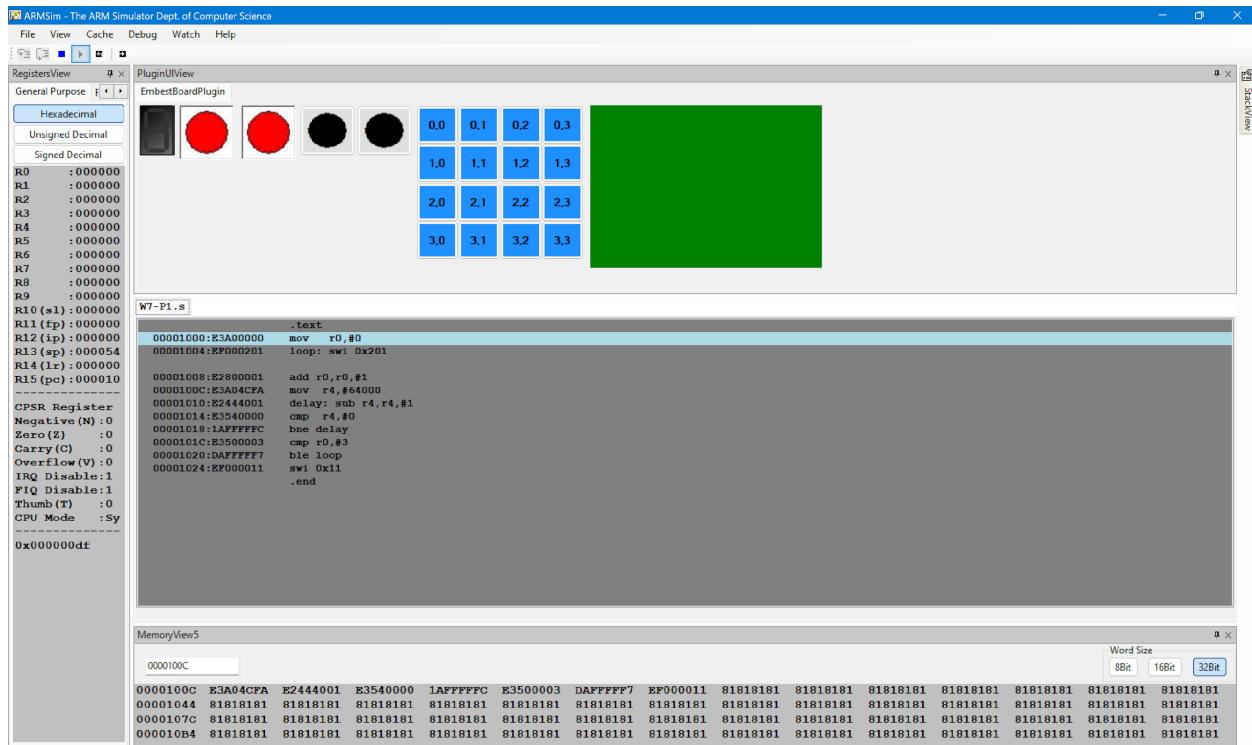
The screenshot shows a dark-themed text editor window titled "W7-P1.s". The menu bar includes "File", "Edit", and "View". The assembly code is as follows:

```
.text
mov r0,#0
loop: swi 0x201

add r0,r0,#1
mov r4,#64000
delay: sub r4,r4,#1
cmp r4,#0
bne delay
cmp r0,#3
ble loop
swi 0x11
.end
```

II. Output Screenshots





Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date: 7/3/2023

Name: Yaman Gupta	SRN:PES2UG21CS619	Section J
-------------------	-------------------	-----------

Week# 7 Program Number: 2

Title of the Program

Write an ALP to generate 0-F on ARMSIM's LED display

I. ARM Assembly Code

II. Output Screen Shots (Two)

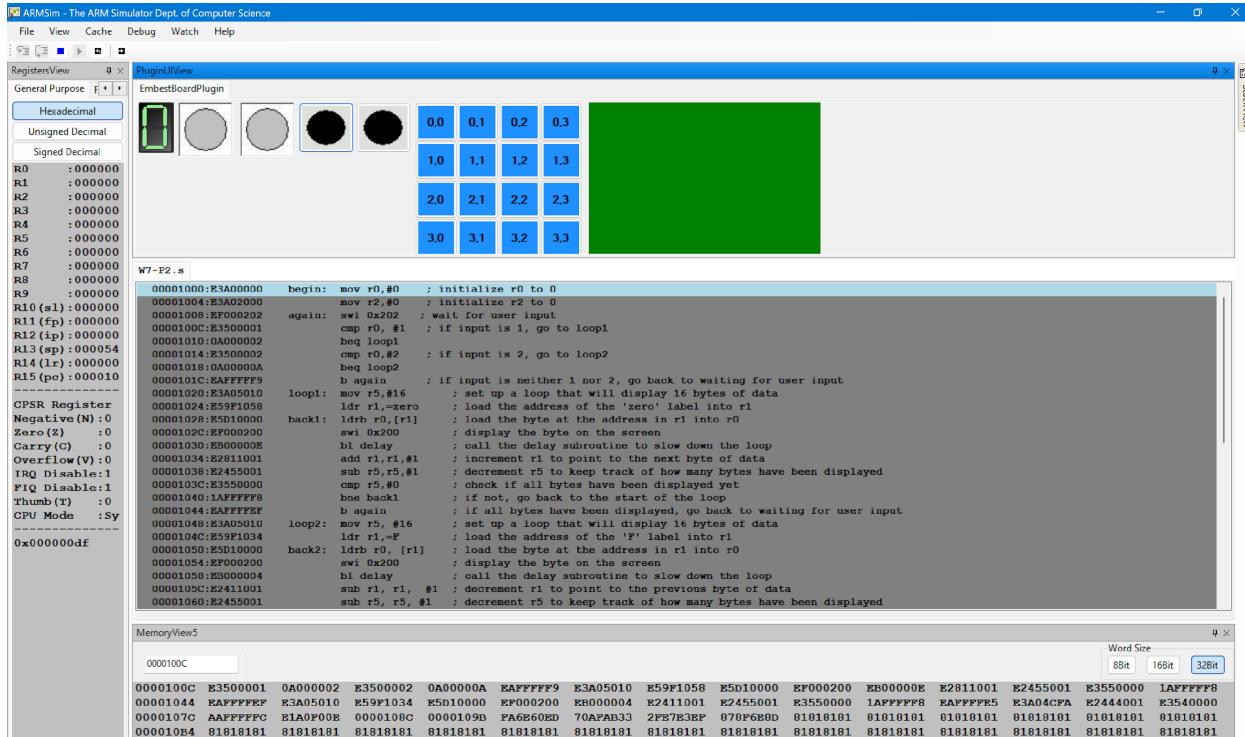
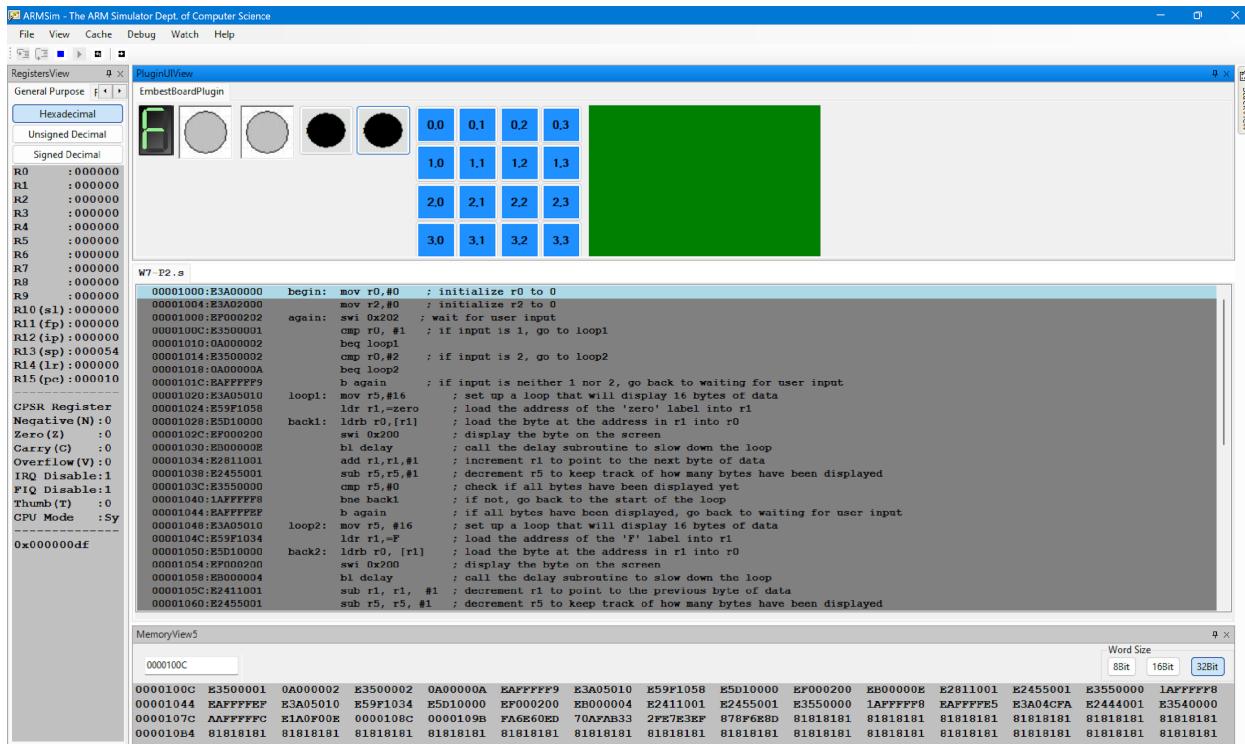
I. ARM Code

The screenshot shows a text editor window with two tabs: "W7-P1.s" and "W7-P2.s". The "W7-P1.s" tab is active and displays the following ARM assembly code:

```
begin:    mov r0,#0      ; initialize r0 to 0
          mov r2,#0      ; initialize r2 to 0
again:    swi 0x202      ; wait for user input
          cmp r0, #1      ; if input is 1, go to loop1
          beq loop1
          cmp r0, #2      ; if input is 2, go to loop2
          beq loop2
          b again        ; if input is neither 1 nor 2, go back to waiting for user input
loop1:   mov r5,#16      ; set up a loop that will display 16 bytes of data
          ldr r1,=zero    ; load the address of the 'zero' label into r1
back1:  ldrb r0,[r1]     ; load the byte at the address in r1 into r0
          swi 0x200      ; display the byte on the screen
          bl delay       ; call the delay subroutine to slow down the loop
          add r1,r1,#1    ; increment r1 to point to the next byte of data
          sub r5,r5,#1    ; decrement r5 to keep track of how many bytes have been displayed
          cmp r5,#0      ; check if all bytes have been displayed yet
          bne back1      ; if not, go back to the start of the loop
          b again        ; if all bytes have been displayed, go back to waiting for user input
loop2:   mov r5, #16      ; set up a loop that will display 16 bytes of data
          ldr r1,=F        ; load the address of the 'F' label into r1
back2:  ldrb r0, [r1]     ; load the byte at the address in r1 into r0
          swi 0x200      ; display the byte on the screen
          bl delay       ; call the delay subroutine to slow down the loop
          sub r1, r1, #1    ; decrement r1 to point to the previous byte of data
          sub r5, r5, #1    ; decrement r5 to keep track of how many bytes have been displayed
          cmp r5,#0      ; check if all bytes have been displayed yet
          bne back2      ; if not, go back to the start of the loop
          b again        ; if all bytes have been displayed, go back to waiting for user input
delay:  mov r4, #64000
loop3: sub r4,r4, #1
cmp r4, #0
bge loop3
mov pc,lr

.data
zero: .byte 0b11101101
one: .byte 0b01100000
two: .byte 0b01101110
three: .byte 0b11110101
four: .byte 0b00110011
five: .byte 0b10101011
six: .byte 0b10101111
seven: .byte 0b01110000
eight: .byte 0b11101111
nine: .byte 0b11100011
A: .byte 0b11100111
B: .byte 0b00101111
C: .byte 0b10001101
D: .byte 0b01101110
E: .byte 0b10001111
F: .byte 0b10000111
```

II. Output Screenshots



Microprocessor and Computer Architecture

UE21CS251B

4th Semester, Academic Year 2022-23

Date:

Name:Yaman Gupta	SRN: PES2UG21CS619	Section J
------------------	-----------------------	--------------

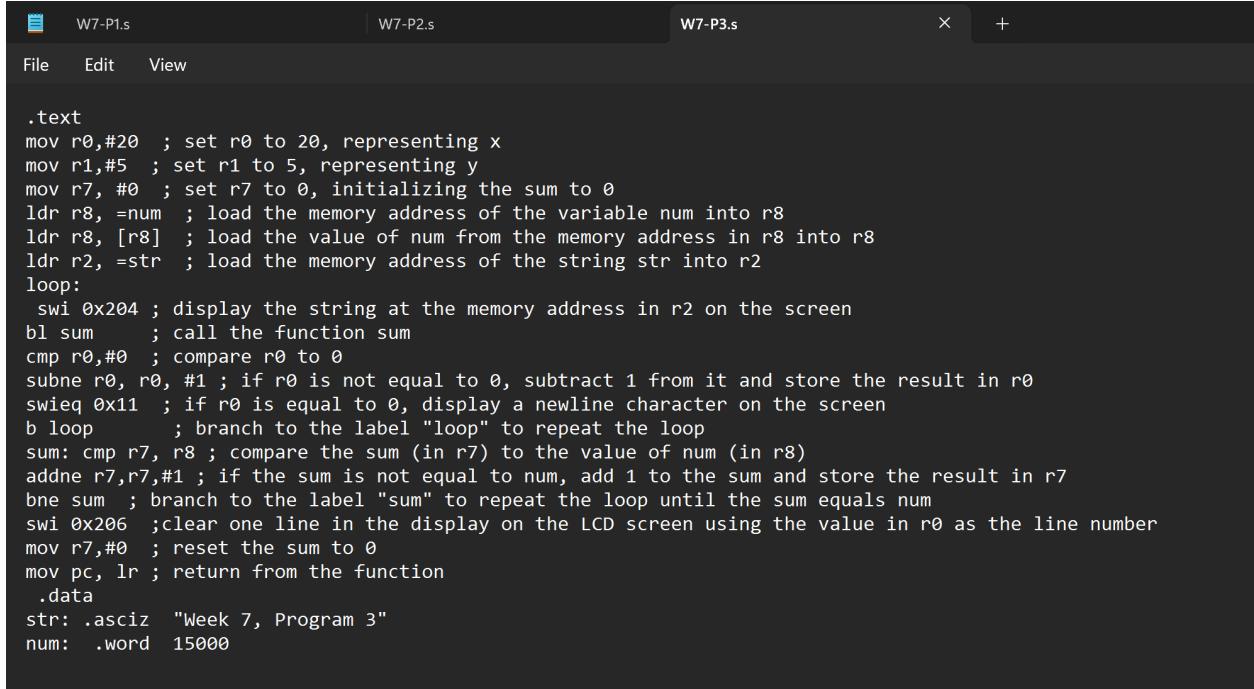
Week# 7 Program Number: 3

Title of the Program

Write an Assembly Level Program to display a String on the LED Display of the Embest Board

- I. Code
- II. Output Screenshots(one)

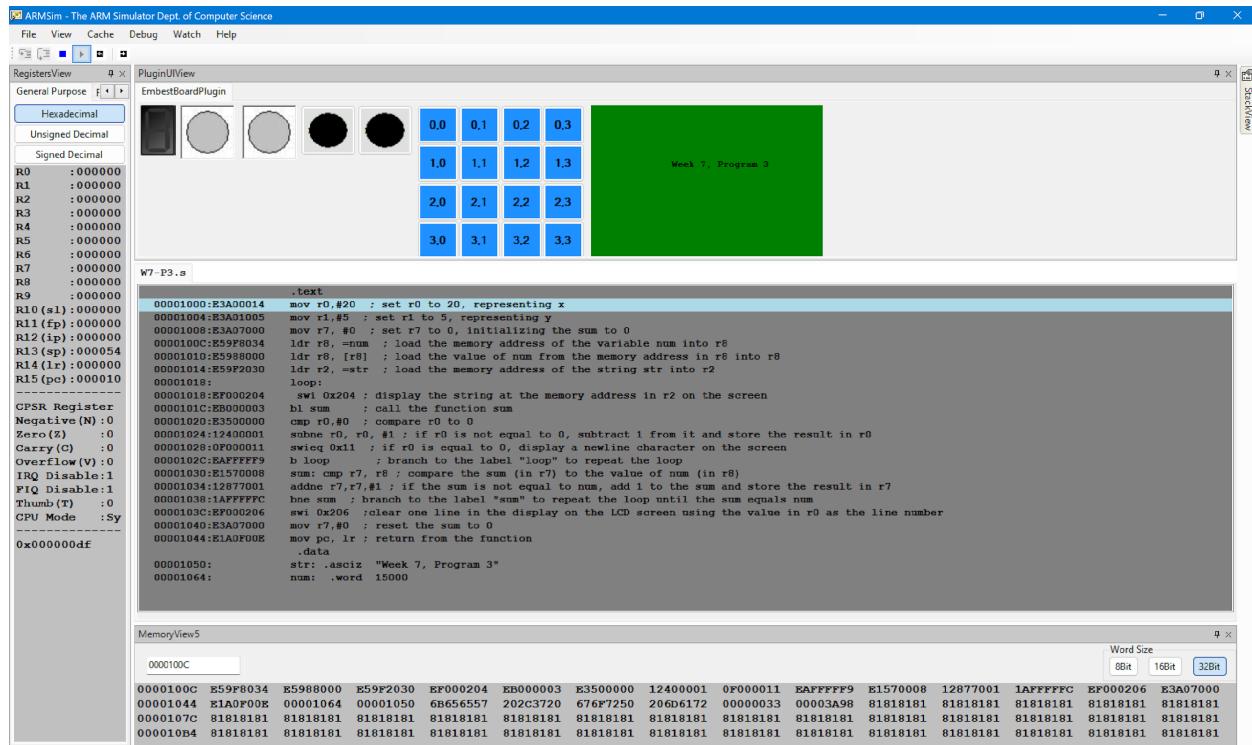
I. Assembly Code



The screenshot shows a terminal window with three tabs at the top: W7-P1.s, W7-P2.s, and W7-P3.s. The W7-P3.s tab is active, displaying the following assembly code:

```
.text
mov r0,#20 ; set r0 to 20, representing x
mov r1,#5 ; set r1 to 5, representing y
mov r7, #0 ; set r7 to 0, initializing the sum to 0
ldr r8, =num ; load the memory address of the variable num into r8
ldr r8, [r8] ; load the value of num from the memory address in r8 into r8
ldr r2, =str ; load the memory address of the string str into r2
loop:
    swi 0x204 ; display the string at the memory address in r2 on the screen
    bl sum ; call the function sum
    cmp r0,#0 ; compare r0 to 0
    subne r0, r0, #1 ; if r0 is not equal to 0, subtract 1 from it and store the result in r0
    swieq 0x11 ; if r0 is equal to 0, display a newline character on the screen
    b loop ; branch to the label "loop" to repeat the loop
sum: cmp r7, r8 ; compare the sum (in r7) to the value of num (in r8)
    addne r7,r7,#1 ; if the sum is not equal to num, add 1 to the sum and store the result in r7
    bne sum ; branch to the label "sum" to repeat the loop until the sum equals num
    swi 0x206 ; clear one line in the display on the LCD screen using the value in r0 as the line number
    mov r7,#0 ; reset the sum to 0
    mov pc, lr ; return from the function
.data
str: .asciz "Week 7, Program 3"
num: .word 15000
```

II. Output Screenshots



Disclaimer:

- The programs and output submitted is duly written, verified and executed by me.
- I have not copied from any of my peers nor from the external resource such as internet.
- If found plagiarized, I will abide with the disciplinary action of the University.

Signature:

Name:Yaman Gupta

SRN: PES2UG21CS619

Section: J

Date: 7/3/23